UNIVERSITY OF WISCONSIN
Computer Sciences Department

CS 537
Fall 2018

Barton Miller

Programming Assignment #1-1

Handed out: Thursday, September 27
Due: Wednesday, October 3 at 5pm

The goals for this assignment are:

A. Learn about tools that can scan your code for style and security mistakes that could make your code more vulnerable. In particular, you'll get experience with the **Clang Static Analyzer (CSA)**.
B. Review the code that you wrote for Program 1 to see what kind of undetected mistakes you might have made.
C. Learn how to understand and fix the mistakes reported.

Your three tasks for this assignment are:

1. Make sure that your Program 1 solution compiles and links ("builds") with `-Wall` and `-Wextra` with **no warnings**. Hopefully, you already did this step when you completed Program 1. If you have not yet done this step, then you'll need to modify your program **to eliminate the warnings**.
2. Run the Clang Static Analyzer on your Program 1 code and figure out what the messages provide by CSA mean. Make changes to your program to eliminate the CSA warnings.
3. Think abouts the results and be ready to discuss in class whether you considered any identified issues seemed to be worth fixing.

If you have questions as to what the messages mean or problems running CSA, talk with a TA or Bart.

Running the Clang Static Analyzer

You can documentation about CSA on the Clang/LLVM website at: https://clang-analyzer.llvm.org/scan-build.html.

There are also lots of other references to be found on CSA on the web.

We're providing you a sample makefile that will run CSA ("make scan-build") and run the the CSA results web viewer ("make scan-view").

You'll need to run this program on one of the first-floor Linux systems, as they have CSA installed. You can also install CSA on your own Linux system.

Re-Testing Your Program

You'll want to make sure that your changes did not affect your program's behavior. So, it should be a simple matter of just re-running your test cases. If you were alert enough to put them in a shell script, it's super easy to do this.

Deliverables

You will work with your original partner for Program 1 and, again, turn in only one copy of your program, including all .c and .h files and your makefile.

Handing in Your Assignment

Your CS537 handin directory is `~cs537-1/handin/your_login` where *your_login* is your CS login. Inside of that directory, you need to create a `proj1-1` subdirectory.

Copying your files to this directory is accomplished with the `cp` program, as follows:

```
    shell% cp *.[ch] makefile README ~cs537-1/handin/your_login/proj1-1
```

You can hand files in multiple times, and later submissions will overwrite your previous ones. To check that your files have been handed in properly, you should list `~cs537-1/handin/your_login/proj1-1` and make sure that your files are there.

When you are working in a pair, you should:

1. Submit only one copy of your code.
2. Make sure that both of your names are in comments at the top of each source file.
3. Both of you should create another file called partner.txt in **both** of your `proj1-1` directories, where you should have two lines that have each of CS login and netid of you and your partner.

Original Work

This assignment must be the original work of you and your project partner. Unless you have explicit permission from Bart, you may not include code from any other source or have anyone else write code for you.

Use of unattributed code is considered plagiarism and will result in academic misconduct proceedings (and "F" in the course and a notation on your transcript).

Last modified: Wed Sep 26 14:01:21 CDT 2018 by bart